Exploring features in a Bayesian framework for material recognition

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td><a href="http://dx.doi.org/10.1109/CVPR.2010.5540207">http://dx.doi.org/10.1109/CVPR.2010.5540207</a></td>
</tr>
<tr>
<td>Publisher</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>Version</td>
<td>Final published version</td>
</tr>
<tr>
<td>Accessed</td>
<td>Tue Oct 23 14:58:02 EDT 2018</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/69562">http://hdl.handle.net/1721.1/69562</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td></td>
</tr>
<tr>
<td>Detailed Terms</td>
<td></td>
</tr>
</tbody>
</table>
Exploring Features in a Bayesian Framework for Material Recognition

Ce Liu\textsuperscript{1,3}, Lavanya Sharan\textsuperscript{2,3}, Edward H. Adelson\textsuperscript{3}, Ruth Rosenholtz\textsuperscript{3}

\textsuperscript{1}Microsoft Research New England  \textsuperscript{2}Disney Research Pittsburgh  \textsuperscript{3}Massachusetts Institute of Technology

celiu@microsoft.com  \{lavanya, adelson, rruth\}@csail.mit.edu

Abstract

We are interested in identifying the material category, e.g. glass, metal, fabric, plastic or wood, from a single image of a surface. Unlike other visual recognition tasks in computer vision, it is difficult to find good, reliable features that can tell material categories apart. Our strategy is to use a rich set of low and mid-level features that capture various aspects of material appearance. We propose an augmented Latent Dirichlet Allocation (aLDA) model to combine these features under a Bayesian generative framework and learn an optimal combination of features. Experimental results show that our system performs material recognition reasonably well on a challenging material database, outperforming state-of-the-art material/texture recognition systems.

1. Introduction

Material recognition is an important aspect of visual recognition. We interact with a variety of materials on a daily basis and we constantly assess their appearance. For example, when judging where to step on an icy sidewalk or buying fresh produce at a farmers’ market or deciding whether a rash requires a trip to the doctor, material qualities influence our decisions. Therefore, it is valuable to build a visual recognition system that can infer material properties from images.

The problem of recognizing materials from photographs has been addressed mainly in the context of reflectance estimation. The visual appearance of a surface depends on several factors – the illumination conditions, the geometric structure of the surface sample at several spatial scales, and the surface reflectance properties, often characterized by the bidirectional reflectance distribution function (BRDF) \cite{24} and its variants \cite{9, 16, 26}. A number of techniques have been developed that can estimate the parameters of a BRDF model from a set of photographs, under restrictive assumptions of illumination, geometry and material properties \cite{10, 11}.

In this paper, we focus on recognizing high-level material categories, such as glass, metal, fabric, plastic or wood, instead of explicitly estimating reflectance properties. The reflectance properties of a material are often correlated with its high-level category (e.g. glass is usually translucent and wood is often brown), and in this work, we will exploit these correlations. However, it is important to point out that knowing only the reflectance properties of a surface is not sufficient for determining the material category. For example, the fact that a surface is translucent does not tell us if it is made of plastic, wax or glass.

Unlike other visual recognition tasks such as object or texture recognition, it is challenging to find good features that can distinguish different material categories because of the wide variations in appearance that a material can display. Our strategy is to design several low-level and middle-level features to characterize various aspects of material appearance. In addition to well-established features such as color, jet and SIFT \cite{17, 21}, we introduce several new features, such as curvature of edges, histogram of oriented gradient (HOG) feature along edges, and HOG perpendicular
Recognizing high-level material categories in images is different from the well-studied problem of object recognition. Although object identity is sometimes predictive of material category, a given class of objects can be made of different materials (see Figure 2) and different classes of objects can be made of the same material (see Figure 1). Therefore, many recent advances in object recognition such as shape context [2], object detectors [7] and label transfer [19] may not be applicable for material recognition. In fact, most object recognition systems rely on material-invariant features and tend to ignore material information altogether.

Material recognition is closely related to, but different from, texture recognition. Texture has been defined in terms of dimensions like periodicity, orientedness, and randomness [20]. It can be an important component of material appearance, e.g. wood tends to have textures distinct from those of polished metal. However, as illustrated in Figure 3, surfaces made of different materials can share the same texture patterns and as a consequence, mechanisms designed for texture recognition [18, 30] may not be ideal for material recognition.

Material recognition is also different from BRDF estimation. In computer graphics, there is great interest in capturing the appearance of real world materials. The visual appearance of materials like wood or skin, has been modeled in terms of the bidirectional reflectance distribution function (BRDF) [10, 22] and related representations such as BTF [9] and BSSRDF [16]. Material recognition might seem trivial if the BRDF is known, but in general, it is nearly impossible to estimate the BRDF from a single image without simplifying assumptions [10, 11].

A number of low-level image features have been developed for identifying materials. The shape of the luminance histogram of images was found to correlate with human judgement of surface albedo [25], and was used to classify images of spheres as shiny, matte, white, grey etc. [11]. Similar statistics were used to estimate the albedo and gloss of stucco-like surfaces [27]. Several techniques have been developed to search for specific materials in real world photographs such as glass [15, 23] or skin [14].

The choice of databases is, often, the key to success in vi-
visual recognition. The CURET database [9] that consists of images of 61 different texture samples under 205 different viewing and lighting conditions, has become the standard for evaluating 3-D texture classification algorithms. A variety of methods based on texton representations [6, 18, 29], bidirectional histograms [8] and image patches [30] have been successful at classifying CURET surfaces (> 95% accuracy). The KTH-TIPS2 database [5] consisting of 11 texture categories, 4 samples per category, and each photographed under a variety of conditions, was introduced to increase the intra-class variation. It was shown that a SVM-based classifier achieves 98.5% accuracy on this database [5]. Our Flickr Materials Database [28] contains 10 material categories and 100 diverse samples in category. On inspecting the images in Figure 1 and the plots in Figure 4, it is apparent that the Flickr Materials Database is more challenging than the CURET database, and for this reason we chose the Flickr Materials Database to develop and evaluate our material recognition system.

3. Features for Material Recognition

In order to build a material recognition system, it is important to identify features that can distinguish material categories from one another. What makes metal look like metal and wood look like wood? Is it color (neutral vs. browns), textures (smooth vs. grainy) or reflectance properties (shiny vs. matte)? Since little is known about which features are suited for material recognition, our approach is to try a variety of features, some borrowed from the fields of object and texture recognition, and some new ones developed specifically for material recognition. From a rendering point of view, once the camera and the object are fixed, the image of the object can be determined by (i) the BRDF of the surface, (ii) surface structures, (iii) object shape and (iv) environment lighting. Given the diversity of appearance in the Flickr Materials Database, we will attempt to incorporate all these factors in our features.

(a) Color and Texture

Color is an important attribute of surfaces and can be a cue for material recognition: wooden objects tend to be brown, leaves are green, fabrics and plastics tend to be saturated with vivid color, whereas stones tend to be less saturated. We extract $3 \times 3$ pixel patches from an RGB image as our color feature.

Texture, both of the wallpaper and 3-D kind [26], can be useful for distinguishing materials. For example, wood and stone have signature textures that can easily tell them apart. We use two sets of features to measure texture. The first set comprises the filter responses of an image through a set of multi-scale, multi-orientation Gabor filters, often called filter banks or jet [17]. Jet features have been used to recognize 3-D textures [18, 30] by clustering to form textons and using the distribution of textons as a feature. The second set of features we use is SIFT [21]. SIFT features have been widely used in scene and object recognition to characterize the spatial and orientational distribution of local gradients [13].

(b) Micro-texture

Two surfaces sharing the same BRDF can look different if they have different surface structures, e.g. if one is smooth and the other is rough. In practice, we usually touch a surface to sense how rough (or smooth) it is. However, our visual system is able to perceive these properties even without a haptic input. For example, we can see tiny hairs on fabric, smooth surfaces in glass objects, crinkles in leather and grains in paper.

In order to extract information about surface structure, we followed the idea in [1], of smoothing an image by bilateral filtering [12] and then using the residual image for further analysis. The process is illustrated in Figure 7. We choose three images from material categories (a) - glass, metal and fabric - and perform bilateral filtering to obtain base image in (b) and display the residual in (d). The residual of bilateral filtering reveals variations in pixel intensity at a finer scale. For the fabric and metal example in Figure 7, the residual is due to surface structure whereas for glass, these variations are related to translucency. Although it is hard to cleanly separate the contributions of surface structure from those of the BRDF, the residual contains useful information about material category. We apply the same approach for characterizing the residual as we did for texture.
Figure 7. Some features for material recognition. From top to bottom is glass, metal and fabric. For an image (a) we apply bilateral filtering [1] to obtain the base image (b). We run Canny edge detector [4] on the base image and obtain edge maps (c). Curvatures of the edges are extracted as features. Subtracting (b) from (a), we get the residual image (d) that shows micro structures of the material. We extract micro-jet and micro-SIFT features on (d) to characterize material micro-surface structure. In (e), we also show some random samples of edge slices along the normal directions of the Canny edges. These samples reveal lighting-dependent features such as specular highlights. The edge ribbon samples are shown in (f). Arrays of HOG’s [7] are extracted from (e) and (f) to form edge-slice and edge-ribbon features.

We compute the jet and SIFT features of the residual image, and name them micro-jet and micro-SIFT for clarity.

(c) Outline Shape

Though a material can be cast into any arbitrary shape, the outline shape of a surface and its material category are often related. e.g. fabrics and glass have long, curved edges, while metals have straight lines and sharp corners. The outline shape of a surface can be captured by an edge map. We run the Canny edge detector [4] on the base image, trim out short edges, and obtain the edge map shown in Figure 7 (c). To characterize the variations in the edge maps across material categories, we measured the curvature on the edge map at three different scales as a feature (see Figure 6).

(d) Reflectance-based features

Glossiness and transparency are important cues for material recognition. Metals are mostly shiny, whereas wooden surfaces are usually dull. Glass and water are translucent, while stones are often opaque. These reflectance properties sometimes manifest as distinctive intensity changes at the edges in an image. To measure these changes, as shown in Figure 6 (b), we extract histogram of oriented gradients (HOG) [7] features along the normal direction of edges. We take a slice of pixels with a certain width along the normal direction, compute the gradient at each pixel, divide the slice into 6 cells, and quantize the oriented gradients in to 12 angular bins. This feature is called edge-slice. We also measure how the images change along the tangent direction of the edges in a similar manner, as suggested in Figure 6 (c). This feature is called edge-ribbon, which is also quantized by 6 cells and 12 angular bins for each cell.

We have described a pool of features that can be potentially useful for material recognition: color, SIFT, jet, micro-SIFT, micro-jet, curvature, edge-slice and edge-ribbon. The flowchart of how our system generates these features is shown in Figure 5. Amongst these features, color, SIFT and jet are low-level features directly computed from the original image and they are often used for texture analysis. The rest of the features, micro-SIFT, micro-jet, curvature, edge-slice and edge-ribbon are mid-level features that rely on estimations of base images and edge maps (Figures 7 (b) & (c)). A priori, we do not know which of these features will perform well. Hence, we designed a Bayesian learning framework to select best combination of features.

4. A Bayesian Computational Framework

Now that we have a pool of features, we want to combine them to build an effective material recognition system. We quantize the features into visual words and extend the LDA [3] framework to select good features and learn per-class distributions for recognition.

4.1. Feature quantization and concatenation

We use the standard k-means algorithm to cluster the instances of each feature to form dictionaries and map image
features into visual words. Suppose there are \( m \) features in
the feature pool and \( m \) corresponding dictionaries \( \{D_i\}_{i=1}^m \).
Each dictionary has \( V_i \) codewords, i.e., \( |D_i| = V_i \). Since
features are quantized separately, the words generated by
the \( i \)th feature are \( \{w_i^{(1)}, \cdots, w_i^{(g_i)}\} \), \( w_i^{(g_i)} \in \{1, 2, \cdots, V_i\} \) and
\( N_i \) is the number of words. In order to put a set of different
words together, a document of \( \{w_i\} \) is composed as:
\[
\{w_1^{(1)}, \cdots, w_1^{(1)}_1, \{w_2^{(1)}, \cdots, w_2^{(2)}\}, \cdots, \{w_m^{(m)}, \cdots, w_m^{(m)}\} \}
\]  

(1)
can be augmented to one set
\[
\{w_1^{(1)}, \cdots, w_1^{(1)}_1, \{w_2^{(1)}, \cdots, w_2^{(2)}+V_i\}, \cdots, \{w_m^{(m)}+\sum_{i=1}^{m-1} V_i, \cdots, w_m^{(m)}+\sum_{i=1}^{m-1} V_i\} \}
\]  

(2)
with a joint dictionary \( \mathbb{D} = \cup_i D_i \). In this way,
we reduced the multi-dictionary problem to a single-dictionary one.

4.2. Latent Dirichlet Allocation

The latent Dirichlet allocation (LDA) [3] was invented
to model the hierarchical structures of words. Details of
the model can be found in [3, 13]. In order to be self-
contained, we will briefly describe the model in the context
of material recognition. As depicted in the graphical
model in Figure 8, we first randomly draw the material
category \( c \sim \text{Mult}(c|\pi) \) where \( \text{Mult}(\cdot|\pi) \) is a multinomial
distribution with parameter \( \pi \). Based on \( c \), we select a
hyper-parameter \( \alpha_c \), based on which we draw \( \theta \sim \text{Dir}(\theta|\alpha_c) \) where
\( \text{Dir}(\cdot|\alpha_c) \) is a Dirichlet distribution with parameter
\( \alpha_c \). \( \theta \) has the following property: \( \sum_{i=1}^{k} \theta_i = 1 \) where \( k \) is
the number of elements in \( \theta \). From \( \theta \) we can draw a series
of topics \( z_n \sim \text{Mult}(z|\theta), n = 1, \cdots, N \). The topic
\( z_n \in \{1, \cdots, k\} \) selects a multinomial distribution \( \beta_{z_n} \) from
which we draw a word \( w_n \sim \text{Mult}(w_n|\beta_{z_n}) \), which corresponds
to a quantization cluster of the features. Unlike
[13] where \( \beta \) is assumed to be a parameter, we impose a
conjugate prior \( \eta \) upon \( \beta \) to account for insufficient data as
suggested by [3].

Since it is intractable to compute the log likelihood
log \( p(w|\alpha_c, \eta) \), we instead maximize the lower bound
\( \mathcal{L}(\alpha_c, \eta) \) estimated through the variational distributions
over \( \theta, \{z_d\}, \beta \). Please refer to [3] for details on deriving the

variational lower-bound and parameter learning for \( \alpha \) and \( \eta \).
Once we have learned \( \alpha_c \) and \( \eta \), we can use Bayesian MAP
criterion to choose the material category
\[
c^* = \arg \max_c \mathcal{L}(\alpha_c, \eta) + \lambda_c. 
\]  

(3)
where \( \lambda_c = \log \pi_c \).

4.3. Prior learning

A uniform distribution is often assumed for the prior
\( p(c) \), i.e., each material category will appear equally. However,
since we learn the LDA model for each category independently (only sharing
the same \( \beta \)), the learning procedure may not converge in finite iterations.
Therefore, the probability density functions (pdfs) should be grounded for a fair
comparison. We designed the following greedy algorithm to
learn \( \lambda \) by maximizing the recognition rate (or minimizing the error).

Suppose \( \{\lambda_i\}_i \neq c \) is fixed and we want to optimize \( \lambda_c \) to
maximize the rate. Let \( q_{d,i} \) be the label for document \( d \). Let
\( q_{d,i} = L_{d}(\alpha_c, \eta) + \lambda_i \) be the “log posterior” for document \( d \)
to belong to category \( i \). Let \( f_d = \max_c q_{d,i} \) be the maximum posterior for
document \( d \). We define two sets:
\[
\Omega_c = \{d | q_{d,c} = \max_c q_{d,i} \},
\Phi_c = \{d | q_{d,c} \neq \max_c q_{d,i} \}.
\]  

(4)
Set \( \Omega_c \) includes the documents that are labeled as \( c \) and
misclassified. Set \( \Phi_c \) includes the documents that are not labeled
as \( c \) and incorrectly classified. Our goal is to choose \( \lambda_c \)
to make \( |\Omega_c| \) as small as possible and \( |\Phi_c| \) as large as possible.
Notice that if we increase \( \lambda_c \), then \( |\Omega_c| \downarrow \) and \( |\Phi_c| \downarrow \),
therefore the optimal \( \lambda_c \) exists. We define the set of correctly
classified documents with \( \lambda_c^* \) as:
\[
\Psi_c = \{d | d \in \Omega_c, f_d > q_{d,c} + \lambda_c^* - \lambda_c \} \cup \\
\{d | d \in \Phi_c, f_d > q_{d,c} + \lambda_c^* - \lambda_c \}.
\]  

(5)
and choose the new \( \lambda_c \) that maximizes the size of \( \Psi_c \):
\[
\lambda_c \leftarrow \arg \max_{\lambda_c} |\Psi_c|.
\]  

(6)
We iterate this procedure for each \( c \) repeatedly until each \( \lambda_c \)
does not change.

4.4. Augmented LDA (aLDA)

Shall we use all the features in our predefined feature
pool? Do more features imply better performance? Unfortunately,
this is not true as we have limited training data. The more
features we use, the more likely that the model
overfits the training data and the performance decreases on
test set. We designed a greedy algorithm in Figure 9 to select
an optimal subset of our feature pool. The main idea is
to select the best feature, one at a time, that maximizes the
recognition rate on an evaluation set. The algorithm stops
when adding more features will decrease the recognition
rate. Note that we randomly split the training set \( H \) into
\( L \) for parameter learning and \( E \) for cross evaluation. After
\( \mathbb{D} \) is learned, we use the entire training set \( H \) to relearn
the parameters for \( \mathbb{D} \).
SIFT color for test. All the experimental results reported in this paper we randomly chose 50 images for training and 50 images sider pixels inside this binary mask for material recognition image describing the location of the object. We only con-
category contains 100 images, 50 of which are close-up views (every
5. Experimental Results

We used the Flickr Materials Database [28] for all experiments described in this paper. There are ten material categories in the database: fabric, foliage, glass, leather, metal, paper, plastic, stone, water and wood. Each category contains 100 images, 50 of which are close-up views and the rest 50 are of views at object-scale (see Figure 1). There is a binary, human-labeled mask associated with each image describing the location of the object. We only consider pixels inside this binary mask for material recognition and disregard all the background pixels. For each category, we randomly chose 50 images for training and 50 images for test. All the experimental results reported in this paper are based on the same split of training and test.

We extract features for each image according to Figure 5. Mindful of computational costs, we sampled color, jet, SIFT, micro-jet and micro-SIFT features on a coarse grid (every 5th pixel in both horizontal and vertical directions). Because there are far fewer pixels in edge maps than in the original images, we sampled every other edge pixel for curvature, edge-slice and edge-ribbon. Once features are extracted, they are clustered separately using k-means according to the number of clusters in Table 1. We specified the number of clusters for each feature, considering both dimensionality and the number of instances per feature.

After forming the dictionaries for each feature, we run the aLDA algorithm to select features incrementally. When learning the optimal feature set, we randomly split the 50 training images per category (set H) to 30 for estimating parameters (set L) and 20 for evaluation (set E). After the feature set is learned, we re-learn the parameters using the
50 training images per category and report the training/test rate. In the LDA learning step, we vary the number of topics from 50 to 250 with step size 50 and pick the best one. The learning procedure is shown in Figure 10, where for each material category we plot the training rate on the left in a darker color and test rate on the right in a lighter color. In Figure 10, the recognition rate is computed on the entire training/test set, not just on the learning/evaluation set. First, the system tries every single feature and discovers that amongst all features, SIFT produces the highest evaluation rate. In the next iteration, the system picks up color from the remaining features, and then edge-slice. Including more features causes the performance to drop and the algorithm in Figure 9 stops. For this final feature set “color + SIFT + edge-slice”, the training rate is 49.4% and the test rate is 44.6%. The recognition rate of random guesses is 10%.

The boost in performance from the single best feature (SIFT, 35.4%) to the best feature set (color + SIFT + edge-slice, 44.6%) is due to our aLDA model that augments visual words. Interestingly, augmenting more features decreases the overall performance. When we use all the features, the test rate is 38.8%, lower than using fewer features. More features creates room for overfitting, and one solution to combat overfitting is to increase the size of the database. The fact that SIFT is the best-performing single feature in-

<table>
<thead>
<tr>
<th>Feature name</th>
<th>Dim</th>
<th>average # per image</th>
<th># of clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color</td>
<td>27</td>
<td>6326.0</td>
<td>150</td>
</tr>
<tr>
<td>Jet</td>
<td>64</td>
<td>6370.0</td>
<td>200</td>
</tr>
<tr>
<td>SIFT</td>
<td>128</td>
<td>6033.4</td>
<td>250</td>
</tr>
<tr>
<td>Micro-jet</td>
<td>64</td>
<td>6370.0</td>
<td>200</td>
</tr>
<tr>
<td>Micro-SIFT</td>
<td>128</td>
<td>6033.4</td>
<td>250</td>
</tr>
<tr>
<td>Curvature</td>
<td>3</td>
<td>3759.8</td>
<td>100</td>
</tr>
<tr>
<td>Edge-slice</td>
<td>72</td>
<td>2461.3</td>
<td>200</td>
</tr>
<tr>
<td>Edge-ribbon</td>
<td>72</td>
<td>3068.6</td>
<td>200</td>
</tr>
</tbody>
</table>

Table 1. The dimension, number of clusters and average number per image for each feature.
Zisserman’s (VZ) algorithm [30] on the Flickr Materials database captured by micro-SIFT. Edge-slice, which measures patches as codewords, obtains a histogram of the codewords. In addition, SIFT also encapsulates some of the information captured by micro-SIFT. Edge-slice, which measures reflectance features, is also useful.

For comparison, we implemented and tested Varma-Zisserman’s (VZ) algorithm [30] on the Flickr Materials Database. The VZ algorithm clusters $5 \times 5$ pixel gray-scale patches as codewords, obtains a histogram of the codewords for each image, and performs recognition using a nearest neighbor classifier. As a sanity check, we ran our implementation of VZ on the CURET database and obtained 96.1% test rate (their numbers are 95 ~ 98%. [30]). Next, we ran the exact VZ system tested on CURET on the Flickr Materials Database. The VZ test rate is 23.8%. This supports the conclusions from Figure 4 that the Flickr Materials Database is much harder than the CURET texture database.

As the VZ system uses features tailored for the CURET database ($5 \times 5$ pixel patches), we ran VZ’s algorithm using our features on Flickr Materials Database. The results of running VZ’s system on exactly the same feature sets as in Figure 10 are listed in Figure 11. Since VZ uses a nearest neighbor classifier, it is meaningless to report the training rate as it is always 100%, so we only report the test rate. It is obvious why many of our features outperform fixed size gray-scale patch features on Flickr Materials Database. In fact, the VZ system running on SIFT features has test rate of 31.8%, close to our system using SIFT alone (35.2%). However, combining features under the VZ’s framework only slightly increases the performance to a maximum of 37.4%. Clearly, the aLDA model contributes to the boost in performance from 37.4% to 44.6%.

The confusion matrix of our system (color + SIFT + edge-slice, test rate 44.6%) in Figure 12 tells us how often each category is misclassified as another. For example, fabric is often misclassified as stone, leather misclassified as fabric, plastic misclassified as paper. The category metal is more likely to be classified as glass than itself. Some misclassification examples are shown in Figure 12. These results are not surprising because there are certain commonalities between leather and fabric, plastic and paper, as well as...
6. Discussion and Conclusion

Although the recognition rate achieved by our system 44.6% is lower than the rates reported in object recognition (e.g. [19]), it is significantly higher than the state of the art (23.8%, [30]). As illustrated in Figures 1 and 4, the sheer diversity and range of the Flickr Materials Database makes it a challenging benchmark for material recognition. We believe that material recognition is an important problem to study, and in this paper, we are merely taking one of the first steps towards understanding the problem.

To conclude, we have presented a set of features and a Bayesian computational framework for material category recognition. Our features were chosen to capture various aspects of material appearance in the real world. An augmented LDA (aLDA) framework was designed to select an optimal set of features by maximizing the recognition rate on the training set. We have demonstrated a significant improvement in performance when using our system over the state of the art on the challenging Flickr Materials Database [28]. We have also analyzed the contribution of each feature in our system to the performance gain. Our feature set and computational framework constitute the first attempt at recognizing high-level material categories “in the wild”.

References